Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1 - 3(Canceled).

4(Currently amended). A system to determine a solutions for a problems including a solution space represented by one or more nodes, said system comprising:

a computer system to dynamically configure and emulate a hardware architecture of a processing system that processes said one or more nodes to determine [[a]] said solution for [[a]] said problem ~~including a solution space represented by one or more nodes,~~ wherein ~~with~~ at least one of said nodes includes ~~including~~ data for said problem~~, wherein~~ and said computer system includes:

an operating system to control operation of said computer system;

a virtual machine unit to emulate said hardware architecture and manage said nodes within said solution space, wherein said hardware architecture is ~~based on processing~~ designed to process said nodes;

a virtual operating system to configure said hardware architecture and to control operation of said virtual machine unit to emulate said hardware architecture ~~in accordance with~~ based on a user software application defining said problem and corresponding processing to determine said solution, said virtual operating system including:

an instantiation engine to create and delete said nodes;

a configuration engine to configure said nodes of said solution space in a topology ~~suitable for determining~~ to determine said solution for said problem;

a population engine to store and evaluate said data for said problem within said nodes;

a navigation engine to traverse said topology and process selected ones of said nodes ~~in accordance with~~ based on said user software application to determine said solution; and

an evolution engine to update said nodes and said topology ~~in accordance with~~ based on said user software application.

5(Previously presented). The system of claim 4, wherein said virtual machine unit includes:

an instantiation unit to create and delete said nodes in response to instructions received from at least one of said instantiation engine and said evolution engine;

a population unit to store and evaluate said data within said nodes in response to instructions received from at least one of said population engine and said evolution engine;

a navigation unit to traverse said topology and retrieve information from said selected nodes in response to instructions received from said navigation engine and said evolution engine; and

a configuration unit to store parameters defining said topology of said nodes representing said problem and configuration parameters for said hardware architecture in response to instructions received from said configuration engine.

6(Previously presented). The system of claim 5, wherein said computer system further includes:

a virtual assembler to convert said instructions from said instantiation, population, navigation and evolution engines into a format compatible with said hardware architecture emulated by said virtual machine unit.

7(Previously presented). The system of claim 4, wherein said computer system further includes:

a platform driver unit to interface said virtual machine unit with said computer system, wherein said platform driver unit converts commands from said emulated hardware architecture into commands compatible with said computer system.

8(Previously presented). The system of claim 4, wherein said virtual operating system includes:

a thread unit to create one or more additional instances of said processing system that each include subsets of said emulated hardware architecture, wherein each instance is associated with a corresponding task to enable said tasks to be performed concurrently.

9(Previously presented). The system of claim 4, wherein said computer system includes an arithmetic logic unit, and said virtual machine unit further includes:

a virtual arithmetic unit to emulate an arithmetic logic unit of said hardware architecture, and to selectively pass arithmetic operations for said emulated hardware architecture to said arithmetic logic unit of said computer system.

10(Previously presented). The system of claim 4, wherein said computer system includes at least one of local and distributed networks of processing systems, and said emulated hardware architecture includes a corresponding instruction set.

11(Previously presented). The system of claim 4, wherein each node includes an index word and a data word.

12(Previously presented). The system of claim 4, wherein each node includes one or more of numeric tags, character tags, boolean flags, numeric values, character values, object identifications, database-record identifications, simple arrays, variable-density multidimensional arrays, symbolic functions, mathematical functions, connection pointers to other nodes, function pointers, lookup-table list pointers, linked-lists, and pointers to other solution spaces, data representations, procedures, or other emulated hardware architectures.

13(Previously presented). The system of claim 4, wherein said topology includes at least one of independent point-clouds, ordered sets of points, acyclic graphs, cyclic graphs, balanced trees, recombining graphs, meshes, and lattices.

14(Previously presented). The system of claim 9, wherein said virtual arithmetic unit provides fixed-point integer arithmetic with precision indicated by said user software application.

15(Previously presented). The system of claim 10, wherein said virtual machine unit includes:

a network unit to manage distribution of data and processes to said networked processing systems.

16(Previously presented). The system of claim 4, wherein said virtual operating system includes:

a process unit to manage daemons for background processing of said nodes; and

a toolbox unit to enable performance of frequently-used tasks.

17(Previously presented). The system of claim 4, wherein said hardware architecture includes a non-Von Neumann architecture.

18(Previously presented). The system of claim 4, wherein said hardware architecture includes a reduced instruction set computer (RISC) architecture.

19(Currently amended). A program product apparatus including a computer useable memory with computer program logic stored therein to enable a computer system with an operating system to dynamically configure and emulate a hardware architecture of a processing system that processes one or more nodes representing a solution space for a problem to determine a solution for [[a]] said problem, wherein ~~said problem includes a solution space represented by one or more nodes with~~ at least one of said nodes includes ~~including~~ data for said problem, said program product apparatus comprising:

a virtual machine unit to emulate said hardware architecture and manage said nodes within said solution space, wherein said hardware architecture is ~~based on processing~~ designed to process said nodes;

a virtual operating system to configure said hardware architecture and to control operation of said virtual machine unit to emulate said hardware architecture ~~in accordance with~~ based on a user software application defining said problem and corresponding processing to determine said solution, said virtual operating system including:

an instantiation engine to create and delete said nodes;

a configuration engine to configure said nodes of said solution space in a topology ~~suitable for determining~~ to determine said solution for said problem;

a population engine to store and evaluate said data for said problem within said nodes;

a navigation engine to traverse said topology and process selected ones of said nodes ~~in accordance with~~ based on said user software application to determine said solution; and

an evolution engine to update said nodes and said topology ~~in accordance with~~ based on said user software application.

20(Previously presented). The apparatus of claim 19, wherein said virtual machine unit includes:

an instantiation unit to create and delete said nodes in response to instructions received from at least one of said instantiation engine and said evolution engine;

a population unit to store and evaluate said data within said nodes in response to instructions received from at least one of said population engine and said evolution engine;

a navigation unit to traverse said topology and retrieve information from said selected nodes in response to instructions received from said navigation engine and said evolution engine; and

a configuration unit to store parameters defining said topology of said nodes representing said problem and configuration parameters for said hardware architecture in response to instructions received from said configuration engine.

21(Previously presented). The apparatus of claim 20, further including:

a virtual assembler to convert said instructions from said instantiation, population, navigation and evolution engines into a format compatible with said hardware architecture emulated by said virtual machine unit.

22(Previously presented). The apparatus of claim 19, further including:

a platform driver unit to interface said virtual machine unit with said computer system, wherein said platform driver converts commands from said emulated hardware architecture into commands compatible with said computer system.

23(Previously presented). The apparatus of claim 19, wherein said virtual operating system includes:

a thread unit to create one or more additional instances of said processing system that each include subsets of said emulated hardware architecture, wherein each instance is associated with a corresponding task to enable said tasks to be performed concurrently.

24(Previously presented). The apparatus of claim 19, wherein said computer system includes an arithmetic logic unit, and said virtual machine unit further includes:

a virtual arithmetic unit to emulate an arithmetic logic unit of said hardware architecture, and to selectively pass arithmetic operations for said emulated hardware architecture to said arithmetic logic unit of said computer system.

25(Previously presented). The apparatus of claim 19, wherein said computer system includes at least one of local and distributed networks of processing systems, and said emulated hardware architecture includes a corresponding instruction set.

26(Previously presented). The apparatus of claim 19, wherein each node includes an index word and a data word.

27(Previously presented). The apparatus of claim 19, wherein each node includes one or more of numeric tags, character tags, boolean flags, numeric values, character values, object identifications, database-record identifications, simple arrays, variable-density multidimensional arrays, symbolic functions, mathematical functions, connection pointers to other nodes, function pointers, lookup-table list pointers, linked-lists, and pointers to other solution spaces, data representations, procedures, or other emulated hardware architectures.

28(Previously presented). The apparatus of claim 19, wherein said topology includes at least one of independent point-clouds, ordered sets of points, acyclic graphs, cyclic graphs, balanced trees, recombining graphs, meshes, and lattices.

29(Previously presented). The apparatus of claim 24, wherein said virtual arithmetic unit provides fixed-point integer arithmetic with precision indicated by said user software application.

30(Previously presented). The apparatus of claim 25, wherein said virtual machine unit includes:

a network unit to manage distribution of data and processes to said networked processing systems.

31(Previously presented). The apparatus of claim 19, wherein said virtual operating system includes:

a process unit to manage daemons for background processing of said nodes; and

a toolbox unit to enable performance of frequently-used tasks.

32(Previously presented). The apparatus of claim 19, wherein said hardware architecture includes a non-Von Neumann architecture.

33(Previously presented). The apparatus of claim 19, wherein said hardware architecture includes a reduced instruction set computer (RISC) architecture.

34(Currently amended). A method of dynamically configuring and emulating a hardware architecture of a processing system that processes one or more nodes representing a solution space for a problem, via a computer system with an operating system, to determine a solution for [[a]] said problem, wherein ~~including a solution space represented by one or more~~

~~nodes with~~ at least one of said nodes includes ~~including~~ data for said problem, said method comprising:

(a) emulating said hardware architecture to implement a virtual machine, via said computer system, and managing said nodes within said solution space, wherein said hardware architecture is ~~based on processing~~ designed to process said nodes;

(b) configuring said hardware architecture, via a virtual operating system, and controlling said emulation of said hardware architecture ~~in accordance with~~ based on a user software application defining said problem and corresponding processing to determine said solution, wherein step (b) further includes:

(b.1) configuring said nodes of said solution space in a topology ~~suitable for determining~~ to determine said solution for said problem and storing and evaluating said data for said problem within said nodes;

(b.2) traversing said topology and processing selected ones of said nodes ~~in accordance with~~ based on said user software application to determine said solution; and

(b.3) updating said nodes and said topology ~~in accordance with~~ based on said user software application.

35(Previously presented). The method of claim 34, wherein step (b.1) further includes:

(b.1) generating and sending instructions to said virtual machine to create and delete said nodes, to store and evaluate said data within said nodes, and to store parameters defining

said topology of said nodes representing said problem and configuration parameters for said hardware architecture;

step (b.2) further includes:

(b.2.1) generating and sending instructions to said virtual machine to traverse said topology and retrieve information from said selected nodes; and

step (b.3) further includes:

(b.3.1) generating and sending instructions to said virtual machine to update said nodes and said topology.

36(Previously presented). The method of claim 35, wherein step (b) further includes:

(b.4) converting said instructions from said virtual operating system into a format compatible with said virtual machine.

37(Previously presented). The method of claim 34, further including:

(c) interfacing said virtual machine with said computer system by converting commands from said virtual machine into commands compatible with said computer system.

38(Previously presented). The method of claim 34, wherein step (a) further includes:

(a.1) creating one or more additional instances of said processing system that each include subsets of said emulated hardware architecture, wherein each instance is associated with a corresponding task to enable said tasks to be performed concurrently.

39(Previously presented). The method of claim 34, wherein said computer system includes an arithmetic logic unit, and step (a) further includes:

(a.1) emulating an arithmetic logic unit of said hardware architecture, and selectively passing arithmetic operations for said emulated hardware architecture to said arithmetic logic unit of said computer system.

40(Previously presented). The method of claim 34, wherein said computer system includes at least one of local and distributed networks of processing systems, and said virtual machine includes a corresponding instruction set.

41(Previously presented). The method of claim 34, wherein each node includes an index word and a data word.

42(Previously presented). The method of claim 34, wherein each node includes one or more of numeric tags, character tags, boolean flags, numeric values, character values, object identifications, database-record identifications, simple arrays, variable-density multidimensional arrays, symbolic functions, mathematical functions, connection pointers to other nodes, function pointers, lookup-table list pointers, linked-lists, and pointers to other solution spaces, data representations, procedures, or other virtual machines.

43(Previously presented). The method of claim 34, wherein said topology includes at least one of independent point-clouds, ordered sets of points, acyclic graphs, cyclic graphs, balanced trees, recombining graphs, meshes, and lattices.

44(Previously presented). The method of claim 39, wherein said emulated arithmetic logic unit provides fixed-point integer arithmetic with precision indicated by said user software application.

45(Previously presented). The method of claim 40, wherein step (b) further includes:

(b.4) managing distribution of data and processes to said networked processing systems.

46(Previously presented). The method of claim 34, wherein step (b) further includes:

(b.4) managing daemons for background processing of said nodes; and

(b.5) enabling performance of frequently-used tasks via a toolbox.

47(Previously presented). The method of claim 34, wherein said hardware architecture includes a non-Von Neumann architecture.

48(Previously presented). The method of claim 34, wherein said hardware architecture includes a reduced instruction set computer (RISC) architecture.

49(Previously presented). The method of claim 34, wherein said emulation of said hardware architecture includes employing at least one of a small instruction set, simple and efficient data representation and handling, inherent vector representation, limited data/calculation modes, interleaved memory, table lookup, induced pointers, and distributed and parallelized computation.

50(Previously presented). The method of claim 34, wherein step (b.2) further includes:

(b.2.1) pre-computing said data within said nodes of said solution space to enable navigation of possible solutions to occur in near real-time.

51(Previously presented). The method of claim 46, wherein said daemons operate concurrently to perform tasks including at least one of collecting garbage, pruning trees, condensing redundancies, processing edit-queues, interpolating with finer granularity around selected nodes in said solution space, and extrapolating and elaborating said data during processing and navigation of said nodes.